Dokumentationsbeschreibung

Da sich unser Projekt über mehrere Dateien und über mehrere Entwicklungsstufen erstreckt, ist eine Beschreibung darüber von Nöten auf welche Art und Weise und in welcher Menge der entstehende und entstandene Code zu dokumentieren ist um anderen Teammitglieder die nicht von Anfang an dabei oder mit anderen Aufgaben betraut waren, den Einstieg in das Projekt zu erleichtern, natürlich sollen auch die Kontrolleure wissen was der Zweck und die Verwendung einer Klasse ist. Wir kommentieren in Deutsch, das erleichtert uns die Arbeit, zu mindest ein bisschen da bei jedem Gruppenmitglied mindestens gute Englischkenntnisse vorhanden sind, da jedoch das Programm im Rahmen eines Projektes der Universität Leipzig entsteht ist eine deutsche Kommentierung am sinnvollsten. Zur Erstellung der Dokumentation verwenden wir JavaDoc, das Dokumentationstool das Java mittlerweile standardmäßig integriert hat, hierzu sind jedoch spezielle Dokumentationen erforderlich um sie mit JavaDoc nutzbar zu machen.

Die Art der Kommentierung geschieht wie folgt:

```
/**
* Dies ist eine beliebige Klasse
*/
class beliebige_Klasse()
        /**
        * Dies ist eine beliebige Methode
        */
        beliebige_Methode()
```

Die Kommentierung auf diese Art verwenden wir besonders bei Klassen, Instanzen, Methoden und Interfaces. Es sollen also alle diese Programmteile so kommentiert werden, dass eine sinnvolle Nutzung von Javadoc möglich wird. Hier die Tags die verwendet werden können :

@see Klassenname: fügt einen 'see also: ' ein, mit einem Link zu der anderen Klasse
auch zu anderen Paket möglich, aber Angabe des Pakets vor dem Klassenname auch zu Methoden in Klassen möglich, dazu muss nach dem Klassennamen eine '#' stehen und dann der Methodenname (@see Klassenname# Methode)
schreibt man den Klassennamen aus, wird er auch mit allen Parametern genauer beschrieben

@param Parametername Beschreibung: fügt eine formale Erklärung des/der Parameter mit Beschreibung ein

@exeception Klassenname Beschreibung: fügt eine Beschreibung einer Execption einer Klasse ein

@throws Klassenname Beschreibung: fügt eine Beschreibung ein wohin geworfen wird

@return Beschreibung: fügt eine Beschreibung des Rückgabewertes ein

@version Text: fügt den 'Text' als Versionsinfo ein, javadoc mit Parameter –version starten

@author Text: fügt den 'Text' als Autor ein, javadoc mit Parameter -author starten

Um die HTML-Dateien in einem anderen Verzeichnis zu erstellen kann man den Parameter '-d Verzeichnis' verwenden, die HTML-Dateien werden dann in diesem Verzeichnis erstellt.
Die Beschreibung zur Einbindung von Tags bezieht sich auf die JavaDoc - Version die der dem Java SDK 1.4.1 beiliegt, deshalb können bei älteren Javaversionen die Aufrufparameter von den hier Angegebenen abweichen.

Des Weiteren sollen auch entscheidende Stellen im Quellcode kommentiert werden.
Dies ist jedoch nur für die Programmierer entscheidend damit eine einfache Navigation und eine einfache Weitergabe an andere Teammitglieder erfolgen kann.
Diese Kommentare sollen den Sinn bzw. die Verwendung des Codes an dieser Stelle darstellen und erklären, gegebenenfalls auch die Ein- und Ausgabe.
Werden komplizierte Codekontruktionen verwendet ist auch eine allgemeine Beschreibung des Codes von Nöten. Dazu gehört ebenfalls eine gute Strukturierung des Codes, also gut einrücken um die Lesbarkeit zu verbessern. Es wird nur eine Anweisung pro Zeile verwendet, nur bei Ausnahmen, z.B. 'counter++;' oder ähnlich kurzen, auch zwei bis drei Anweisungen pro Zeile, jedoch maximal 80 Zeichen pro Zeile. Dies dient dazu den Code auch neben der Anweisung kurz kommentieren zu können. Leerzeilen werden zwischen Klassen/Interface- Deklarationen, zwischen Methoden, zwischen lokalen Variablendeklarationen und zwischen semantisch abgeschlossenen Quelltextabschnitten gesetzt.
Paketnamen werden komplett klein geschrieben, bei Klassen und Interfaces wird der erste Buchstabe und jedes neue Wort groß geschrieben, bei Methoden wird der erste Buchstabe klein geschrieben und soweit möglich mit Verben beschrieben, z.B. getYear, Variablen und Attribute werden auch klein angefangen und dann bei neuem Wort groß fortgesetzt, dabei sollten außer bei temporalen Variablen keine Abkürzungen verwendet werden und Klassenkonstanten werden komplett groß geschrieben.
Zur besseren Lesbarkeit werden auch ausreichend Leerzeichen gesetzt, bei Formeln: (a + b), und auch bei Schleifen : while (i > 0) und nach Kommas. Variablennamen sind so zu wählen, dass der Sinn und die Verwendung der Variable spätestens auf den zweiten Blick zu erkennen ist. Soll also beispielsweise ein Zähler verwendet werden ist ein Variablenname wie 'counter' oder 'zaehler' oder ähnliches zu verwenden. Das gleiche gilt natürlich auch für die Namen der einzelnen Java bzw. Class Dateien.

Auch einen kurzen Kommentar vor Klassen anderen oben beschrieben Programmteilen um sich schnell einen Überblick über den Nutzen des folgenden Codes ist angebracht. Dies erleichtert die Suche in verschiedenen Dokumenten wenn eine bestimmte Programmstelle gesucht wird. Das heißt mindestens eine einleitende Kommentarzeile vor neuen Klassen, Methoden und Konstruktoren die nicht in JavaDoc übernommen wird.